

International Journal on Artificial Intelligence Tools  
 © World Scientific Publishing Company

## HOW TO USE TEMPORAL-DRIVEN CONSTRAINED CLUSTERING TO DETECT TYPICAL EVOLUTIONS

MARIAN-ANDREI RIZOIU

*ERIC Laboratory, University Lumière Lyon 2  
 5, avenue Pierre Mendès France  
 69676 Bron Cedex, France  
 Marian-Andrei@rizoiu.eu*

JULIEN VELCIN

*ERIC Laboratory, University Lumière Lyon 2  
 Julien.Velcin@univ-lyon2.fr*

STÉPHANE LALLICH

*ERIC Laboratory, University Lumière Lyon 2  
 Stephane.Lallich@univ-lyon2.fr*

Received (Day Month Year)

Revised (Day Month Year)

Accepted (Day Month Year)

In this paper, we propose a new time-aware dissimilarity measure that takes into account the temporal dimension. Observations that are close in the description space, but distant in time are considered as dissimilar. We also propose a method to enforce the segmentation contiguity, by introducing, in the objective function, a penalty term inspired from the Normal Distribution Function. We combine the two propositions into a novel time-driven constrained clustering algorithm, called **TDCK-Means**, which creates a partition of coherent clusters, both in the multidimensional space and in the temporal space. This algorithm uses soft semi-supervised constraints, to encourage adjacent observations belonging to the same entity to be assigned to the same cluster. We apply our algorithm to a Political Studies dataset in order to detect typical evolution phases. We adapt the Shannon entropy in order to measure the entity contiguity, and we show that our proposition consistently improves temporal cohesion of clusters, without any significant loss in the multidimensional variance.

*Keywords:* semi-supervised clustering; temporal clustering; temporal-aware dissimilarity measure; contiguity penalty function; temporal cluster graph structure.

### 1. Introduction

Researchers in Social Sciences and Humanities (like Political Studies) have always gathered data and compiled databases of knowledge. This information often has a temporal component, the evolution of a certain number of entities is recorded over a period of time. Each entity is described using multiple attributes, which form the multidimensional description space. Therefore, an entry in such a database would be an observation, a triple  $(entity, timestamp, description)$ . An observation  $x_i = (\phi_l, t_m, x_i^d)$  signifies that the en-

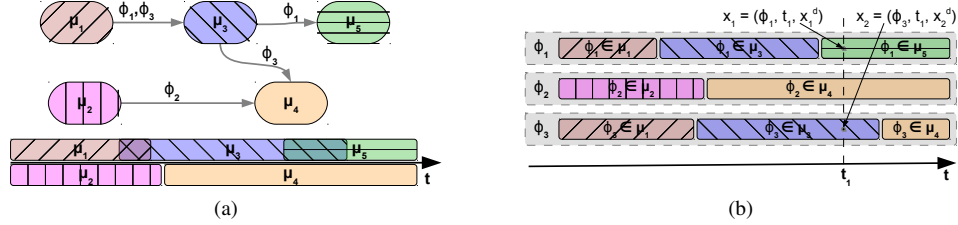
2 *Rizoiu et al.*

Fig. 1. Desired output: (a) the evolution phases and the entity trajectories, (b) the observations of 3 entities contiguously partitioned into 5 clusters.

tity  $\phi_l$  is described by the vector  $x_i^d$  at the moment of time  $t_m$ . We denote by  $x_i^\phi$  the entity to which the observation  $x_i$  is associated. Similarly,  $x_i^t$  is the timestamp associated with the observation  $x_i$ . Each observation belongs to a single entity and, consequently, each entity is associated with multiple observations, for different moments of time. Formally:

$$\forall x_i \in \mathcal{D} : \exists! \phi_l \in \Phi \text{ so that } x_i^\phi = \phi_l$$

$$\forall (\phi_l, t_m) \in \Phi \times \mathcal{T} : \exists! x_i = (x_i^\phi, x_i^t, x_i^d) \text{ so that } x_i^\phi = \phi_l \text{ and } x_i^t = t_m$$

For example, a database studying the evolution of democratic states<sup>1</sup> will store, for each country and each year, the value of multiple economical, social, political and financial indicators. The countries are the entities, and the years are the timestamps.

Starting from such a database, one of the interests of Political Studies researchers is to detect typical evolution patterns. There is a double interest: a) obtaining a broader understanding of the phases that the entity collection went through over time (*e.g.* detecting the periods of global political instability, of economic crisis, of wealthiness *etc.*); b) constructing the trajectory of an entity through the different phases (*e.g.* a country may have gone through a period of military dictatorship, followed by a period of wealthy democracy). The criteria describing each phase are not known beforehand (which indicators announce a world economic crisis?) and may differ from one phase to another.

We address these issues by proposing a novel temporal-driven constrained clustering algorithm. The proposed algorithm partitions the observations into clusters  $\mu_j \in \mathcal{M}$ , that are coherent both in the multidimensional description space and in the temporal space. We consider that the obtained clusters can be used to represent the typical phases of the evolution of the entities through time. Figure 1 shows the desired result of our clustering algorithm. Each of the three depicted entities ( $\phi_1, \phi_2$  and  $\phi_3$ ) is described at 10 moments of time ( $t_m, m = 1, 2, \dots, 10$ ). The 30 observations of the dataset are partitioned into 5 clusters ( $\mu_j, j = 1, 2, \dots, 5$ ). In Figure 1a we observe how clusters  $\mu_j$  are organized in time. Each of the clusters has a limited extent in time, and the time extents of clusters can overlap. The temporal extent of a cluster is the minimal interval of time that contains all the timestamps of the observations in that cluster. The entities navigate through clusters. When an observation belonging to an entity is assigned to cluster  $\mu_2$  and the anterior observation of the same entity is assigned in cluster  $\mu_1$ , then we consider that the entity has a transition from phase  $\mu_1$  to phase  $\mu_2$ . Figure 1b shows how the series of observations belonging to each entity are assigned to clusters, thus forming continuous segments. This succession of segments is

interpreted as the succession of phases through which the entity passes. For this succession to be meaningful, each entity should be assigned to a rather limited number of continuous segments. Passing through too many phases reduces the comprehension. Similarly, evolutions which are alternations between two phases (*e.g.*,  $\mu_1 \longrightarrow \mu_2 \longrightarrow \mu_1 \longrightarrow \mu_2$ ) hinder the comprehension.

Based on these observations, we assume that the resulting partition must:

- **regroup observations having similar descriptions into the same cluster** (just as traditional clustering does). The clusters represent a certain type of evolution;
- **create temporally coherent clusters, with limited extent in time.** In order for a cluster to be meaningful, it should regroup observations which are temporally close (be contiguous on the temporal dimension). If there are two different periods with similar evolutions (*e.g.* two economical crises), it is preferable to have them regrouped separately, as they represent two distinct phases. Furthermore, while it is acceptable that some evolutions exist during the entire period, usually the resulted clusters should have a limited temporal extent;
- **segment, as contiguously as possible, the series of observations for each entity.** The sequence of segments will be interpreted as the sequence of phases through which the entity passes.

In this paper, we propose a new time-aware dissimilarity measure that takes into account the temporal dimension. Observations that are close in the description space, but distant in time are considered as dissimilar. We also propose a method to enforce the segmentation contiguity, by introducing a penalty term based on the Normal Distribution Function. We combine the two propositions into a novel time-driven constrained clustering algorithm, **TDCK-Means**, which creates a partition of coherent clusters, both in the multidimensional space and in the temporal space. This algorithm uses soft semi-supervised constraints to encourage adjacent observations belonging to the same entity to be assigned to the same cluster. The proposed algorithm constructs the clusters that serve as evolution phases and segments the observations series for each entity.

The paper is organized as follows. In Section 2 we present some previous related works and, in Section 3, we introduce the temporal-aware dissimilarity function, the contiguity penalty, function the TDCK-Means algorithm and the graph structure induction method. In Section 4, we present the dataset that we use, the proposed evaluation measures and the obtained results. Finally, in Section 5, we draw the conclusion and plan some future extensions.

## 2. Related work

Leveraging partial expert knowledge into clustering represents the domain of semi-supervised clustering. The expert knowledge is under the form of either class labels, or pairwise constraints. Pairwise constraints<sup>17</sup> are either “must-link” (the observations must be placed in the same cluster) or “cannot-link” (the two observations cannot be placed in the same cluster). Depending on the method in which supervision is introduced into clustering,<sup>9</sup> divides the semi-supervised clustering methods into two classes: a) the similarity-adapting methods<sup>3,5,11,19</sup>, which seek to learn new similarity measures in order to satisfy

the constraints, and b) the search-based methods<sup>2,7,17</sup> in which the clustering algorithm itself is modified.

The literature presents some examples of algorithms used to segment a series of observations into continuous chunks. In<sup>12</sup>, the daily tasks of a user are detected by segmenting scenes from the recordings of his activities. Semi-supervised must-link constraints are set between all pairs of observations, and a fixed penalty is inflicted when the following conditions are fulfilled simultaneously: the observations are not assigned to the same cluster and the time difference between their timestamps is less than a certain threshold. A similar technique is used in<sup>6</sup>, where constraints are used to penalize non-smooth changes (over time) on the assigned clusters. This segmenting technique is used to detect tasks performed during a day, based on video, on sound and on GPS information. In<sup>15</sup>, the objects appearing in an image sequence are detected by using a hierarchical descending clustering, that regroups pixels into large temporally coherent clusters. This method seeks to maximize the cluster size, while guaranteeing intra-cluster temporal consistency. All of these techniques consider only one series of observations (a single entity) and must be adapted for the case of multiple series. The main problem of a threshold based penalty function is to set the value of the threshold, which is usually data-dependent. Optimal matching is used in<sup>18</sup> to discover trajectory models, while studying the de-standardization of typical life courses.

The temporal dimension of the data is also used in some other fields of Information Retrieval. In<sup>16</sup>, constrained clustering is used to scope temporal relational facts in the knowledge bases, by exploiting temporal containment, alignment, succession, and mutual exclusion constraints among facts. In<sup>4</sup>, clustering is used to segment temporal observations into continuous chunks, as a preprocessing phase. A graphical model is proposed in<sup>14</sup>, that uses a probabilistic model in which the timestamp is part of the observed variables, and the story is the hidden variable to be inferred. But still, none of these approaches seek to create temporally coherent partitions of the data, mainly using the temporal dimension as a secondary information.

In the following sections, we propose a dissimilarity measure, a penalty function and a clustering algorithm in which the temporal dimension has a central role, and which address the limitations existing in the above presented work.

### 3. Temporal-Driven Constrained Clustering

The observations  $x_i \in \mathcal{X}$  that need to be structured can be written as triples (*entity, time, description*):  $x_i = (x_i^\phi, x_i^t, x_i^d)$ .  $x_i^d \in \mathcal{D}$  is the vector in the multidimensional description space which describes the entity  $x_i^\phi \in \Phi$  at the moment of time  $x_i^t \in \mathcal{T}$ .

Traditional clustering algorithms input a set of multidimensional vectors, which they regroup in such a way that observations inside a group resemble each other as much as possible, and resemble observations in other groups as little as possible. K-Means<sup>13</sup> is a clustering algorithm based on iterative relocation, that partitions a dataset into  $m$  clusters, locally minimizing the sum of distances between each data points  $x_i$  and its assigned cluster centroids  $\mu_j \in \mathcal{M}$ . At each iteration, the objective function

$$I = \sum_{\mu_j \in \mathcal{M}} \sum_{x_i \in \mathcal{C}_j} \|x_i^d - \mu_j^d\|^2$$

is minimized until it reaches a local optimum.

Such a system is appropriate for constructing partitions based solely on  $x_i^d$ , the description in the multidimensional space. It does not take into account the temporal order of the observations, nor the structure of the dataset, the fact that observations belong to entities. We extend to the temporal case by adding to the centroids a temporal dimension  $\mu_j^t$ , described in the same temporal space  $\mathcal{T}$  as the observations. Just like its multidimensional description vector  $\mu_j^d$ , the temporal component does not necessary need to exist in the temporal set of the observation. It is an abstraction of the temporal information in the group, serving as a cluster timestamp. Therefore, a centroid  $\mu_j$  will be the couple  $(\mu_j^t, \mu_j^d)$ .

We propose to adapt the K-Means algorithm to the temporal case by adapting the Euclidean distance, normally used to measure the distance between an element and its centroid. This novel temporal-aware dissimilarity measure takes into account both the distance in the multidimensional space and in the temporal space. In order to ensure the temporal contiguity of observations for the entities, we add a penalty whenever two observations that belong to the same entity are assigned to different clusters. The penalty depends on the time difference between the two: the lower the difference, the higher the penalty. We integrate both into the **Temporal-Driven Constrained K-Means (TDCK-Means)**, which is a temporal extension of K-Means. TDCK-Means searches to minimize the following objective function:

$$\mathcal{J} = \sum_{\mu_j \in \mathcal{M}} \sum_{x_i \in \mathcal{C}_j} \left( \|x_i - \mu_j\|_{TA} + \sum_{\substack{x_k \notin \mathcal{C}_j \\ x_k^\phi = x_i^\phi}} w(x_i, x_k) \right) \quad (2)$$

where  $\|\bullet\|_{TA}$  is our temporal-aware (TA) dissimilarity measure (detailed in the next section),  $w(x_i, x_j)$  is the cost function that determines the penalty of clustering adjacent observations of the same entity into different clusters, and  $\mathcal{C}_j$  is the set of observations in cluster  $j$ .

### 3.1. The temporal-aware dissimilarity measure

The proposed temporal-aware dissimilarity measure  $\|x_i - x_j\|_{TA}$  combines the Euclidean distance in the multidimensional space  $\mathcal{D}$  and the distance between the timestamps. We propose to use the following formula:

$$\|x_i - x_j\|_{TA} = 1 - \left( 1 - \frac{\|x_i^d - x_j^d\|^2}{\Delta x_{max}^2} \right) \left( 1 - \frac{\|x_i^t - x_j^t\|^2}{\Delta t_{max}^2} \right) \quad (3)$$

where  $\|\bullet\|$  is the classical  $L^2$  norm and  $\Delta x_{max}$  and  $\Delta t_{max}$  are the diameters of  $\mathcal{D}$ , and  $\mathcal{T}$  respectively (the largest distance encountered between two observations in the multidimensional description space and, respectively, in the temporal space). The following properties are immediate:

- $\|x_i - x_j\|_{TA} \in [0, 1], \forall x_i, x_j \in \mathcal{X}$
- $\|x_i - x_j\|_{TA} = 0 \Leftrightarrow x_i^d = x_j^d \text{ and } x_i^t = x_j^t$
- $\|x_i - x_j\|_{TA} = 1(\text{maximum}) \Leftrightarrow \|x_i^d - x_j^d\| = \Delta x_{max} \text{ or } \|x_i^t - x_j^t\| = \Delta t_{max}$

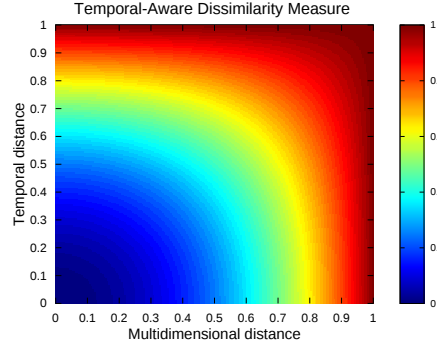


Fig. 2. Color map of the temporal-aware dissimilarity measure as a function of the multidimensional component and the temporal component.

Figure 2 plots the temporal-aware dissimilarity measure as a color map, depending on the multidimensional component and the temporal component. The horizontal axis represents the normalized multidimensional distance ( $\frac{\|x_i^d - x_j^d\|^2}{\Delta x_{max}^2}$ ). The vertical axis represents the normalized temporal distance ( $\frac{\|x_i^t - x_j^t\|^2}{\Delta t_{max}^2}$ ). The blue color shows a temporal-aware measure close to the minimum and the red color represents the maximum. The dissimilarity measure is zero if and only if the two observations have equal timestamps and equal multidimensional description vectors. Still, it suffices for only one of the components (temporal, multidimensional) to attend the maximum value for the measure to reach its maximum. The measure behaves similar to a MAX operator, always choosing a value closer to the maximum of the two components. The formula for the temporal-aware dissimilarity measure was chosen so that any algorithm that seeks to minimize an objective function based on this measure, will need to minimize both its components. This makes it suitable for algorithms that search to minimize both the multidimensional and the temporal variance in clusters.

Both components that intervene in the measure follow a function like  $1 - \epsilon^2$ ,  $\epsilon \in [0, 1]$ . This function provides a good compromise: it is tolerant for small values of  $\epsilon$  (small time difference, small multidimensional distance), but decreases rapidly when  $\epsilon$  augments. The temporal-aware dissimilarity measure is an extension of the Euclidean function. If the timestamps are unknown and set to be all equal, the temporal component is canceled and the temporal-aware dissimilarity measure becomes a normalized Euclidean distance. In Section 4.4, we evaluate the behavior of the proposed dissimilarity function. We will call **Temporal-Driven K-Means** the algorithm that is based on the K-Means' iterative structure and uses the temporal-aware dissimilarity measure to assess similarity between observations. Note that **Temporal-Driven K-Means**, relative to TDCK-Means, has no contiguous segmentation penalty function (the contiguous segmentation penalty function is detailed in the next section).

### 3.2. The contiguity penalty function

The penalty function encourages temporally adjacent observations of the same entity to be assigned to the same cluster. We use the notion of *soft pair-wise constraints* from semi-supervised clustering. A “must-link” soft constraint is added between all pairs of obser-

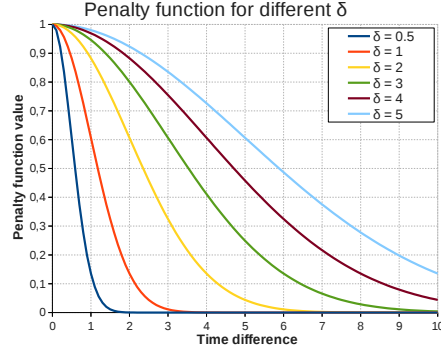


Fig. 3. Penalty function vs. time difference for multiple  $\delta$ . ( $\beta = 1$ )

variations belonging to the same entity. The clustering is allowed to break the constraints, while inflicting a penalty for each of these violations. The penalty is more severe if the observations are closer in time. The function is defined as:

$$w(x_i, x_k) = \beta * e^{-\frac{1}{2} \left( \frac{\|x_i^t - x_k^t\|}{\delta} \right)^2} \mathbb{1} [x_i^\phi = x_k^\phi] \quad (4)$$

where  $\beta$  is a scaling factor and, at the same time, the maximum value taken by the penalty function;  $\delta$  is a parameter which controls the width of the function.  $\beta$  is dataset dependent and can be set as a percentage of the average distance between observations.  $\mathbb{1} [statement]$  is a function that returns 1 if *statement* is true and 0 otherwise.

The function resembles to the positive side of the Normal Distribution function, centered in zero. The function has a particular shape, as represented in Figure 3. For small time differences, it descends very slowly, thus inflicting a high penalty for breaking a constraint. As the time difference increases, the penalty decreases rapidly, converging towards zero. When  $\delta$  is small, the functions value descends very quickly with the time difference. The function produces penalties only if the constraint is broken for adjacent observation. For high values of  $\delta$ , breaking constraints for distant observations cause high penalties, therefore creating segmentations with large segments. Figure 3 shows the evolution of the penalty function with the time difference between two observations, for multiple values of  $\delta$  and for  $\beta = 1$ .

An advantage of the proposed function is that it requires no time discretization or setting a fixed window width, as proposed in <sup>12</sup>. The  $\delta$  parameter permits the fine tuning of the penalty function. In Section 4.4, we evaluate **Constrained K-Means**, which is an extension of K-Means, to which we add the proposed contiguity penalty function (but which does not take into account the temporal dimension when measuring the distance between observations). The influence of both  $\beta$  and  $\delta$  will be studied in Section 4.5.

### 3.3. The TDCK-Means algorithm

The time dependent distance  $\|x_i - \mu_j\|_{TA}$  encourages the decrease of both the temporal and multidimensional variance of clusters; meanwhile the penalty function  $w(x_i, x_k)$  favors the adjacent observations belonging to the same entity to be assigned to the same cluster. The rest of the TDCK-Means algorithm is similar to the K-Means algorithm. It

seeks to minimize  $\mathcal{J}$  by iterating an assignment phase and a centroid update phase until the partition does not change between two iterations. The outline of the algorithm is given in Algorithm 1.

The **choose\_random** function chooses randomly, for each centroid  $\mu_j$ , an observation  $x_i$  and sets  $\mu_j = (x_i^t, x_i^d)$ . In the assignment phase, for every observation  $x_i$ , the **best\_cluster** function chooses a cluster  $\mathcal{C}_j$  so that the temporal-aware dissimilarity measure from  $x_i$  to the cluster's centroid  $\mu_j$ , added to the cost of penalties possibly incurred by this cluster assignment, is minimized. It resumes to solving the following equation:

$$\text{best\_cluster}(i) = \underset{j=1,2,\dots,m}{\operatorname{argmin}} \left( \|x_i - \mu_j^{(iter-1)}\|_{TA}^2 + \sum_{\substack{x_k^\phi = x_i^\phi \\ x_k \notin \mathcal{C}_j^{(iter-1)}}} w(x_i, x_k) \right)$$

This guaranties that the contribution of  $x_i$  to the value of  $\mathcal{J}$  diminishes or stays constant. Overall, this assures that  $\mathcal{J}$  diminishes in the assignment phase (or stays constant).

In the centroid update phase, the **update\_centroid** function recalculates the cluster centroids using the observations in  $\mathcal{X}$  and the assignment at the previous iteration. Therefore the contribution of each cluster to the  $\mathcal{J}$  function is minimized. Each of the temporal and the multidimensional components is calculated individually. In order to find the values that minimize the objective function, we need to solve the equations:

$$\frac{\partial \mathcal{J}}{\partial \mu_j^d} = 0; \quad \frac{\partial \mathcal{J}}{\partial \mu_j^t} = 0 \quad (6)$$

By replacing equations (3) and (4) in (2), we obtain the following formula for the objective function:

$$\begin{aligned} \mathcal{J} = |\mathcal{X}| - \sum_{j=1}^m \sum_{x_i \in \mathcal{C}_j} \left[ \left( 1 - \frac{\|x_i^d - \mu_j^d\|^2}{\Delta x_{max}^2} \right) \left( 1 - \frac{\|x_i^t - \mu_j^t\|^2}{\Delta t_{max}^2} \right) \right] \\ + \sum_{x_i \in \mathcal{X}} \sum_{x_k \notin \mathcal{C}_j} \beta * e^{-\frac{1}{2} \left( \frac{\|x_i^t - x_k^t\|}{\delta} \right)^2} \mathbb{1} [x_i^\phi = x_k^\phi] \end{aligned} \quad (7)$$

Therefore, from equations (6) and (7), we obtain the centroid update formulas:

$$\mu_j^d = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^d \times \left( 1 - \frac{\|x_i^t - \mu_j^t\|^2}{\Delta t_{max}^2} \right)}{\sum_{x_i \in \mathcal{C}_j} \left( 1 - \frac{\|x_i^t - \mu_j^t\|^2}{\Delta t_{max}^2} \right)}; \quad \mu_j^t = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^t \times \left( 1 - \frac{\|x_i^d - \mu_j^d\|^2}{\Delta x_{max}^2} \right)}{\sum_{x_i \in \mathcal{C}_j} \left( 1 - \frac{\|x_i^d - \mu_j^d\|^2}{\Delta x_{max}^2} \right)} \quad (8)$$

Just like the centroid update phase in K-Means, the new centroids in TDCK-Means are also averages over the observations. Unlike K-Means, the averages are weighted for each component, using the distance from the other. For example, each observation contributes to the multidimensional description of the new centroid, proportional with its temporal centrality in the cluster. Observations that are more distant in time (from the centroid) contribute less to the multidimensional description than the ones being closer in time. A similar logic applies to the temporal component. The consequence is that the new clusters are coherent both in the multidimensional space and in the temporal one.



---

**Algorithm 1** Outline of the TDCK-Means algorithm.

---

**Input:**  $x_i \in \mathcal{X}$  - observations to cluster;  
**Input:**  $m$  - number of requested clusters;  
**Output:**  $\mathcal{C}_j, j = 1, 2, \dots, m$  -  $m$  clusters;  
**Output:**  $\mu_j, j = 1, 2, \dots, m$  - centroids for each cluster;

```

for  $j = 1, 2, \dots, m$  do
     $\mu_j \leftarrow \text{choose\_random}(\mathcal{X})$ 
end for
 $iter \leftarrow 0$ 
 $\mathcal{M}^{(iter)} \leftarrow \emptyset$  //set of centroids
 $\mathcal{P}^{(iter)} \leftarrow \emptyset$  //set of clusters
repeat
     $iter \leftarrow iter + 1$ 
    for  $j = 1, 2, \dots, m$  do
         $\mathcal{C}_j^{(iter)} \leftarrow \emptyset$ 
    end for
    // assignment phase
    for  $x_i \in \mathcal{X}$  do
         $\mathcal{C}_j^{(iter)} = \mathcal{C}_j^{(iter)} \cup x_i$  where  $j = \text{best\_cluster}(\mathcal{X}, \mathcal{M}^{(iter-1)}, \mathcal{P}^{(iter-1)})$ 
    end for
    // centroids update phase
    for  $j = 1, 2, \dots, m$  do
         $(\mu_j^{\phi, (iter)}, \mu_j^{t, (iter)}) \leftarrow \text{update\_centroid}(j, \mathcal{X}, \mathcal{M}^{(iter-1)}, \mathcal{P}^{(iter-1)})$ 
    end for
     $\mathcal{M}^{(iter)} \leftarrow \{\mu_j^{(iter)} | j = 1, 2, \dots, m\}$ 
     $\mathcal{P}^{(iter)} \leftarrow \{\mathcal{C}_j^{(iter)} | j = 1, 2, \dots, m\}$ 
until  $\mathcal{C}_j^{(iter)} = \mathcal{C}_j^{(iter-1)}, \forall j \in [1, m]$ 

```

---

**Algorithm's complexity** Equation (7) shows that TDCK-Means' complexity is  $\mathcal{O}(n^2m)$ , due to the penalty term. Still, the equation can be rewritten, so that only observations belonging to the same entity are tested. If  $p$  is the number of entities and  $q$  is the maximum number of observations associated with each entity, then  $n = p \times q$ . The complexity of TDCK-Means is  $\mathcal{O}(pq^2m)$ , which is well adapted to Social Science and Humanities datasets, where often a large number of individuals is studied over a relatively short period of time ( $p > q$ ).

### 3.4. Fine-tuning the ratio between components

The temporal-aware dissimilarity measure, as presented in Equation (3), gives equal importance to both the multidimensional component and the temporal component. This might pose problems when the data are not uniformly distributed both in the multidimensional descriptive space and in the temporal space. If the medium standard deviation reported to the medium distance between pairs of observations is greater in one space than in the other,

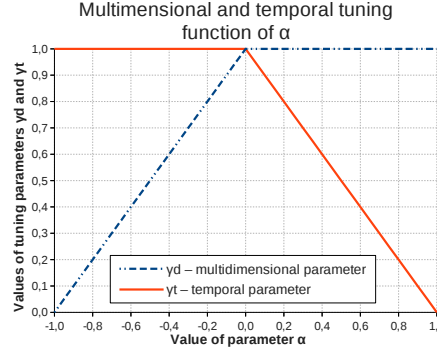


Fig. 4. Multidimensional component, temporal component and temporal-aware dissimilarity measure function of  $\alpha$

giving equal weight to the components can lead to important bias in the clustering process. *E.g.* observations that are very uniformly distributed in the temporal space (same number of observations for each timestamp) and, at the same time, rather compactly distributed in the description space. In this case, in average, the temporal component weight more in the dissimilarity measure than the multidimensional component. Consequently, the clustering is biased towards the temporal cohesion of clusters. Similarly, in some applications, it is desirable to privilege one component over the other. *E.g.* on a large enough scale, user roles in social networks have a temporal component (new types of roles might appear over the years). But in a limited time span, it is perfectly acceptable that the roles can coexist simultaneously. Therefore, the temporal component should have only a mild impact on the overall measure.

We adjust the ratio between the two components by using two tuning factors  $\gamma_d$  and  $\gamma_t$ .  $\gamma_d$  weights the multidimensional component of the temporal-aware dissimilarity measure, whereas  $\gamma_t$  weights the temporal component. Equation (3) can be rewritten as:

$$\|x_i - x_j\|_{TA} = 1 - \left(1 - \gamma_d \frac{\|x_i^d - x_j^d\|^2}{\Delta x_{max}^2}\right) \left(1 - \gamma_t \frac{\|x_i^t - x_j^t\|^2}{\Delta t_{max}^2}\right) \quad (9)$$

When the tuning factor for a certain component is set at zero, the respective component does not contribute to the temporal-aware measure. When the tuning factor is set to one, no penalty is inflicted to the contribution of the respective component to the measure. It is immediate that equation (3) is a special case of equation (9), with  $\gamma_d = 1$  and  $\gamma_t = 1$  (no weights).

**Setting the weights  $\gamma_d$  and  $\gamma_t$**   $\gamma_d$  and  $\gamma_t$  are not independent one from another, their values are set using a unique parameter  $\alpha$ .

$$\gamma_d = \begin{cases} 1 + \alpha, & \text{if } \alpha \leq 0 \\ 1, & \text{if } \alpha > 0 \end{cases} ; \gamma_t = \begin{cases} 1, & \text{if } \alpha \leq 0 \\ 1 - \alpha, & \text{if } \alpha > 0 \end{cases} \quad (10)$$

$\alpha$  acts as a slider, taking values from  $-1$  to  $1$ . Figure 4 shows the evolution  $\gamma_d$  and  $\gamma_t$  with  $\alpha$ . Also, Figure 5 shows the color map of the temporal-aware dissimilarity measure for multiple values of  $\alpha$ .

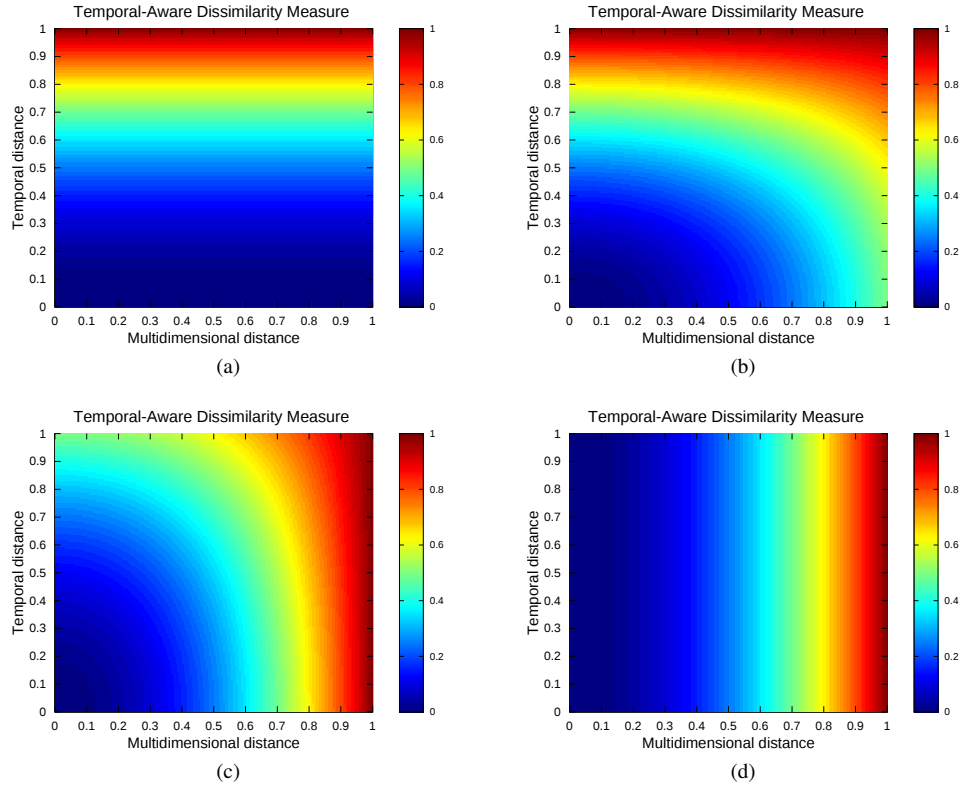


Fig. 5. Color map of the temporal-aware dissimilarity measure for  $\alpha = -1$  (a),  $\alpha = -0.5$  (b),  $\alpha = 0.5$  (c) and  $\alpha = 1$  (d).

When  $\alpha = -1$ , then  $\gamma_d = 0$  and  $\gamma_t = 1$ . The multidimensional component is eliminated and only the time difference between the two observations is considered. The temporal-aware measure becomes a normalized time difference ( $\|x_i - x_j\|_{TA} = \frac{\|x_i^t - x_j^t\|^2}{\Delta t_{max}^2}$ ). The color map in Figure 5a ( $\alpha = -1$ ) shows that the values of the dissimilarity measure is independent of the multidimensional component.

As the value of  $\alpha$  increases, the weight of the descriptive component increases as well. In Figure 5b ( $\alpha = -0.5$ ), the multidimensional component has a limited impact on the overall measure. When  $\alpha = 0$ , then  $\gamma_d = 1$  and  $\gamma_t = 1$ , both components have equal importance, as proposed initially in Equation (3). In Figure 5c ( $\alpha = 0.5$ ), the color map shows that the multidimensional component has a larger impact than the temporal component. Large values of the temporal component have only moderate influence over the measure. When  $\alpha = 1$  (color map in Figure 5d), then  $\gamma_d = 1$  and  $\gamma_t = 0$ , the temporal dimension is eliminated and the measure becomes a normalized Euclidean distance ( $\|x_i - x_j\|_{TA} = \frac{\|x_i^d - x_j^d\|^2}{\Delta x_{max}^2}$ ).

Since the temporal-aware dissimilarity measure is used in the objective function in Equation (7), the later changes accordingly to integrate the tuning factors.  $\gamma_d$  and  $\gamma_t$  behave as constants in the derivation formulas in Equation (6). As a result, the centroid update

formulas in Equation (8) are rewritten as:

$$\mu_j^d = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^d \times \left(1 - \gamma_t \frac{\|x_i^t - \mu_j^t\|^2}{\Delta t_{max}^2}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \gamma_t \frac{\|x_i^t - \mu_j^t\|^2}{\Delta t_{max}^2}\right)} ; \mu_j^t = \frac{\sum_{x_i \in \mathcal{C}_j} x_i^t \times \left(1 - \gamma_d \frac{\|x_i^d - \mu_j^d\|^2}{\Delta x_{max}^2}\right)}{\sum_{x_i \in \mathcal{C}_j} \left(1 - \gamma_d \frac{\|x_i^d - \mu_j^d\|^2}{\Delta x_{max}^2}\right)}$$

The tuning between the multidimensional and temporal component in the temporal-aware dissimilarity measure propagates into the centroid update formula of TDCK-Means. We study, in Section 4.6, the influence of the tuning parameter and we propose an heuristic to set its value.

### 3.5. Inferring a graph structure for the temporal clusters

In Figure 1a, when discussing the desired output of our system, we presented the obtained temporal clusters under the form of a graph. The nodes represent the evolution phases and an edge between two nodes  $\mu_p$  and  $\mu_q$  indicates that the transition  $\mu_p \rightarrow \mu_q$  is part of a typical evolution. Since each temporal cluster is interpreted as an evolution phase, the visualization under the form of a graph allows quick understanding of how the different phases are organized both (i) in time (phases on the left side of Figure 1a have a lower timestamp than those on the right side) and (ii) considering the transitions of the entities through phases. Intuitively, the strength of the connection between two phases is proportional with the number of entities which present transitions between the two given phases.

We consider that an entity  $\phi_l$  presents a transition between  $\mu_p$  and  $\mu_q$  ( $\mu_p \xrightarrow{\phi_l} \mu_q$ ) if and only if two consecutive observations exist, associated with the given entity, where the first observation (ordered by their timestamp) is clustered under  $\mu_p$  and the second one is clustered under  $\mu_q$ . Formally:

$$\begin{aligned} x_a, x_b \in \mathcal{D} \text{ consecutive} &\Leftrightarrow x_a^\phi = x_b^\phi = \phi_l, x_a^t \leq x_b^t \text{ and} \\ &\nexists x_c \in \mathcal{D}, x_c^\phi = \phi_l \text{ so that } x_a^t \leq x_c^t \leq x_b^t \\ \mu_p \xrightarrow{\phi_l} \mu_q &\Leftrightarrow \exists x_a, x_b \in \mathcal{D} \text{ so that } \begin{cases} x_a^\phi = x_b^\phi = \phi_l \text{ and} \\ x_a, x_b \text{ consecutive and} \\ x_a \in \mathcal{C}_p, x_b \in \mathcal{C}_q \end{cases} \end{aligned} \quad (12)$$

Furthermore, we define the intersection similarity measure between two phases, which is based on the normalized number of entities that present a transition between the two phases. Formally, we define the  $inter_\phi(\mu_p, \mu_q)$ :

$$inter_\phi(\mu_p, \mu_q) = \frac{|\{\phi_l \in \Phi | \mu_p \xrightarrow{\phi_l} \mu_q\}|}{|\Phi|} \quad (13)$$

where  $inter_\phi(\mu_p, \mu_q) \in [0, 1]$  and needs to be maximized.

We infer a graph structure between the temporal clusters, by constructing an adjacency matrix using the intersection similarity measure. The graph construction is performed *a posteriori*, after the temporal clusters are calculated. We define the adjacency matrix  $A =$

$(a_{p,q})$ , where  $a_{p,q} = \text{inter}_\phi(\mu_p, \mu_q)$ . By replacing equations 12 and 13 into this definition, we obtain:

$$a_{p,q} = \frac{|\{1 \leq l \leq |\Phi| \mid \exists x_a, x_b \in (\mathcal{C}_p, \mathcal{C}_q) \text{ so that } x_a^\phi = x_b^\phi = \phi_l \text{ and } x_a^t, x_b^t \text{ consecutive}\}|}{|\Phi|} \quad (14)$$

We construct  $A^*$ , a binary adjacency matrix, by using a threshold  $\gamma$ :

$$A^* = (a_{p,q}^*) \text{ with } a_{p,q}^* = \begin{cases} 0 & , \text{ if } a_{p,q} < \gamma \\ 1 & , \text{ if } a_{p,q} \geq \gamma \end{cases}$$

The filtering parameter  $\gamma$  is dataset dependent and automatically setting its value is part of the perspectives of our work.

## 4. Experiments

### 4.1. Dataset

Experimentations with Time-Driven Constrained K-Means are performed on a dataset issued from political sciences: *Comparative Political Data Set I*<sup>1</sup>. It is a collection of political and institutional data, which consists of annual data for 23 democratic countries for the period from 1960 to 2009. The dataset contains 207 political, demographic, social and economic variables.

The dataset was cleaned by removing redundant variables (*e.g.* country identifier and postal code) and the corpus was preprocessed by removing entity bias from the data. For example, it is difficult to compare, on the raw data, the evolution of population between populous country and one with fewer inhabitants, since any evolution in the 50 years timespan of the dataset will be rendered meaningless by the initial difference. Inspired from panel data econometrics<sup>8</sup>, we remove the entity-specific, time-invariant effects, since we assume them to be fixed over time. We subtract from each value the average over each attribute and over each entity. We retain the time-variant component, which is in turn normalized, in order to avoid giving too much importance to certain variables. The obtained dataset is under the form of triples (*country, year, description*).

### 4.2. Qualitative evaluation

When studying the evolution of countries over the years, it is quite obvious for the human reader why the evolutions of the eastern European countries resemble each other for most of the second half of the twentieth century. The reader would create a group entitled “Communism”, extending from right after the Second World War until roughly 1990, for defining the typical evolution of communist countries. One would expect that, based on a political dataset, the algorithms would succeed in identifying such typical evolutions and segment the time series of each of these countries accordingly. Figure 6 shows the typical evolution patterns constructed by TDCK-Means (with  $\beta = 0.003$  and  $\delta = 3$ , obtained as shows in Section 4.5), when asked for 8 clusters. The distribution over time of observations in each cluster is given in Figure 6a. All constructed clusters are fairly compact in time and have limited temporal extents. They can be divided into two temporal groups. In the first one, clusters  $\mu_1$  to  $\mu_5$  consistently overlap. Same for clusters  $\mu_6$  to  $\mu_8$ , in the second group. This indicates that the evolution of each country passes by at least one cluster from each

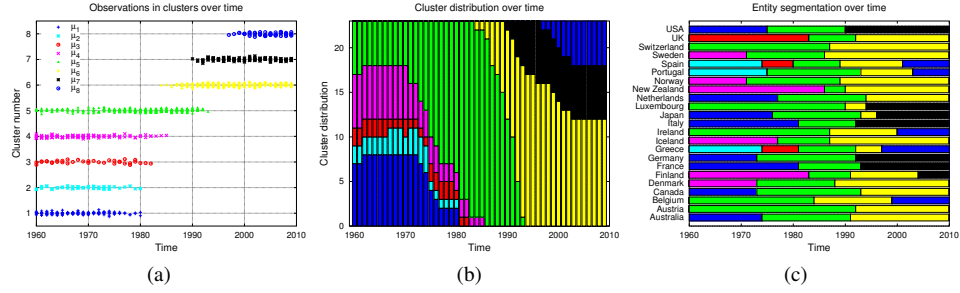


Fig. 6. Typical evolution patterns constructed by TDCK-Means on *Comparative Political Data Set I* with 8 clusters. The distribution over time of observations in each cluster (a), how many entities belong in a certain clusters for each year (b) and the segmentation of entities over clusters (c).

group. The turning point between the two groups is around 1990. Figure 6b shows how many entities belong in a certain cluster for each year. Clusters  $\mu_5$  and  $\mu_6$  contain most of the observations, suggesting the general typical evolution.

The meaning of each constructed cluster starts to unravel only when studying the segmentation of countries over clusters, in Figure 6c. For example, cluster  $\mu_2$  regroups the observations belonging to Spain, Portugal and Greece from 1960 up until around 1975. Historically, this coincides with the non-democratic regimes in those countries (Franco's dictatorship in Spain, the "Regime of the Colonels" in Greece). Likewise, cluster  $\mu_4$  contains observations of countries like Denmark, Finland, Iceland, Norway, Sweden and New Zealand. This cluster can be interpreted as the "Swedish Social and Economical Model" of the Nordic countries, to which the algorithm added, interestingly enough, New Zealand. In the second period, cluster  $\mu_8$  regroups observations of Greece, Ireland, Spain, Portugal and Belgium, the countries which seemed the most fragile in the aftermaths of the economical crises of 2008.

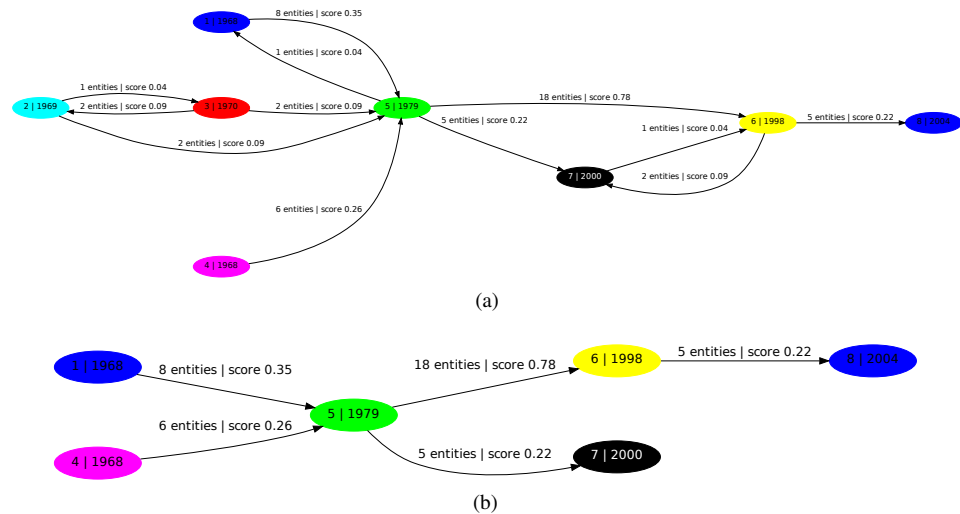


Fig. 7. Structuring the temporal clusters as a graph, without filtering ( $\gamma = 0$ ) (a) and filtered with  $\gamma = 0.2$  (b).

Similar conclusions can be drawn from the constructed graph structure, presented in Figure 7. Each temporal cluster is represented as a node and the scores indicated on each edge are calculated as shown in Equation 14. The graph containing all transitions are represented in Figure 7a (no filtering,  $\gamma = 0$ ). We obtain a graph containing more general evolutions, by filtering with the threshold  $\gamma = 0.2$ . Some “rare” phases completely disappear (*i.e.*, phases  $\mu_2$  and  $\mu_3$ ) together with some of the arcs. We recognize in the resulted graph, shown in Figure 7b, some of the evolutions identified earlier. The evolution  $\mu_4 \rightarrow \mu_5 \rightarrow \mu_6$  corresponds to the “Swedish Social and Economical Model”, whereas the evolution  $\mu_5 \rightarrow \mu_6 \rightarrow \mu_8$  identifies the fragile European economies of the 2008 economical crises. From the filtered evolution graph, another typical evolution emerges:  $\mu_1 \rightarrow \mu_5 \rightarrow \mu_7$ , which is present for countries as USA, Germany, Italy and France. We interpret this evolution as that of countries with stable social and economical environments.

#### 4.3. Evaluation measures

Since the dataset contains no labels to report to as ground truth, we use the classical Information Theory measures in order to numerically evaluate the proposed algorithms. We evaluate separately each of the three goals that we propose in Section 1.

**Create clusters that are coherent in the multidimensional description space.** It is desirable that observations that have similar multidimensional descriptions to be partitioned under the same cluster. The similarity in the description space is measured by the multi-dimensional component of the temporal-aware dissimilarity measure. This goal is pursued by all classical clustering algorithms (like K-Means) and any traditional clustering evaluation measure<sup>10</sup> can be used to asses it. We choose the mean cluster variance, which is traditionally used in clustering to quantify the dispersion of observations in clusters. The *MDvar* measure is defined as:

$$MDvar = \frac{1}{|\mathcal{X}|} \times \sum_{j=1}^m \sum_{x_i \in \mathcal{C}_j} \|x_i^d - \mu_j^d\|^2$$

**Create temporally coherent clusters, with limited extend in time.** This goal is very similar to the previous one, translated in the temporal space. It is desirable that observations that are assigned to the same cluster to be similar in the temporal space (*i.e.* to be close in time). The similarity in the temporal space is measured by the temporal component in the temporal-aware dissimilarity measure. The limited time extent of a centroid implies small temporal distances between observations timestamp and the centroid timestamp. As a result, the variance can also be used to measure the dispersion of clusters in the temporal space. Similarly to *MDvar*, the *Tvar* measure is defined as:

$$Tvar = \frac{1}{|\mathcal{X}|} \times \sum_{j=1}^m \sum_{x_i \in \mathcal{C}_j} \|x_i^t - \mu_j^t\|^2$$

**Segment the temporal series of observations of each entity into a relatively small number of contiguous segments.** The goal is to have successive observations belonging to an entity grouped together, rather that scattered in different clusters. The Shannon entropy can quantify the number of clusters which regroup the observations of an entity, but it is insensible to alternations between two classes (evolutions like  $\mu_1 \rightarrow \mu_2 \rightarrow \mu_1 \rightarrow$

$\mu_2$ ). We evaluate using an adapted mean Shannon entropy of clusters over entities, which weights the entropy by a penalty factor depending on the number of continuous segments in the series of each entity. The *ShaP* measure is calculated as:

$$ShaP = \frac{1}{|\mathcal{X}|} \times \sum_{x_i \in \mathcal{X}} \sum_{j=1}^m \left( -p(\mu_j) \times \log_2(p(\mu_j)) \times \left( 1 + \frac{n_{ch} - n_{min}}{n_{obs} - 1} \right) \right)$$

where  $n_{ch}$  is the number of changes in the cluster assignment series of an entity,  $n_{min}$  is the minimal required number of changes and  $n_{obs}$  is the number of observations for an entity. For example, in Figure 8, if the series of 11 observations of an entity is assigned to two clusters, but it presents 4 changes, the entropy penalty factor will be  $1 + \frac{4-1}{11-1} = 1.33$ . The *ShaP* score for this segmentation will be 1.23, compared to a score of 0.94 of the “ideal” segmentation (only two contiguous chunks).

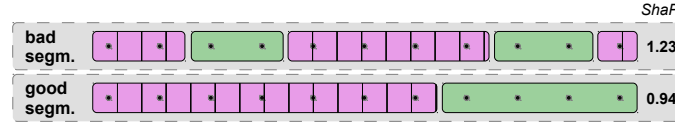


Fig. 8. Examples of a good and a bad segmentation in contiguous chunks and their related *ShaP* score.

The “ideal” values for *MDvar*, *Tvar* and *ShaP* is zero and, in all of the experiments presented in the following sections, we search to minimize the values of the three measures.

#### 4.4. Quantitative evaluation

For each combination of algorithms and parameters, we execute 10 times and compute only the average and the standard deviation. We vary  $m$ , the number of clusters, from 2 to 36. The performances of five algorithms are compared from a quantitative point of view:

- **Simple K-Means** - clusters the observations based solely on their resemblance in the multidimensional space;
- **Temporal-Driven K-Means** - optimizes only the temporal and multidimensional components, without any contiguity constraints; combines K-Means with the temporal-aware dissimilarity measure defined in Section 3.1. Parameters:  $\alpha = 0$  ( $\alpha$  defined in Equation 8) and  $\beta = 0$  ( $\beta$  defined in Equation 4);
- **Constrained K-Means** - uses only the multidimensional space (and not the temporal component) together with the penalty component, as proposed in Section 3.2. Parameters:  $\alpha = 1$ ,  $\beta = 0.003$  and  $\delta = 3$  ( $\delta$  defined in Equation 4);
- **TDCK-Means** - the Temporal-Driven Constrained Clustering algorithm proposed in Section 3.3.  $\alpha = 0$ ,  $\beta = 0.003$  and  $\delta = 3$ ;
- **tcK-Means** - the temporal constrained clustering algorithm proposed in <sup>12</sup>. It uses a threshold penalty function  $w(x_i^{t_i}, x_j^{t_i}) = \alpha^* \mathbb{1}(|x_i^t - x_j^t| < d^*)$  when observations  $x_i$  and  $x_j$  are not assigned to the same cluster. We adapt it to the multi-entity case by applying it only to observations belonging to the same entity. Parameters:  $\alpha^* = 2$ ,  $d^* = 4$ .

The  $\alpha^*$  parameter in **tcK-Means** should not be mistaken with the  $\alpha$  parameter in **TDCK-Means**, as they do not have the same meaning. In **tcK-Means**,  $\alpha^*$  controls the



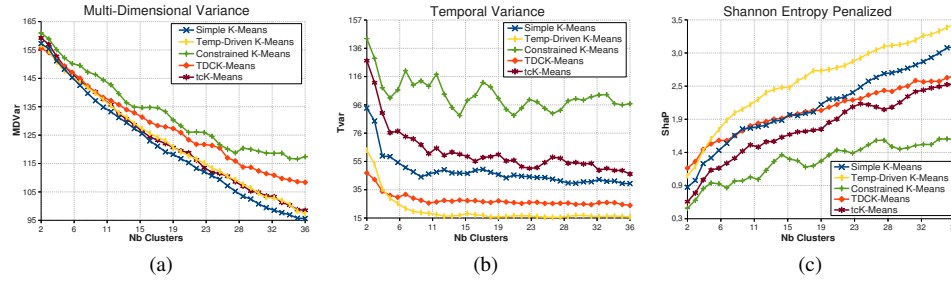
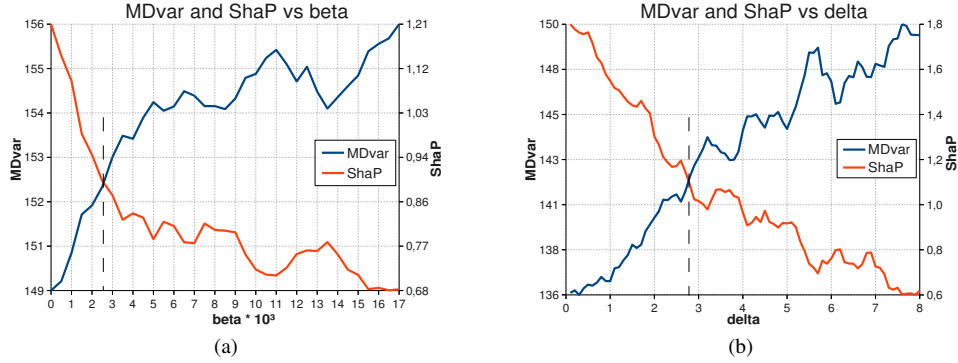


Fig. 9. *MDvar* (a), *Tvar* (b) and *ShaP* (c) values of the considered algorithms when varying the number of clusters.

Algorithm		<i>MDvar</i>		<i>Tvar</i>		<i>ShaP</i>	
Scores	<b>Simple K-Means</b>	<b>120.59</b>	2.97	48.01	8.87	2.15	0.23
	<b>Temp-Driven K-Means</b>	122.98	2.85	<b>19.97</b>	5.39	2.58	0.18
	<b>Constrained K-Means</b>	132.69	8.07	103.15	42.98	<b>1.24</b>	0.5
	<b>TDCK-Means</b>	127.81	3.96	27.54	5.81	2.06	0.2
	<b>tcK-Means</b>	123.04	3.8	62.44	24.16	1.79	0.32
% Gain	<b>Temp-Driven K-Means</b>	-1.99%		<b>58.40%</b>		-19.63%	
	<b>Constrained K-Means</b>	-10.04%		-114.84%		<b>42.21%</b>	
	<b>TDCK-Means</b>	-5.99%		42.64%		4.19%	
	<b>tcK-Means</b>	-2.03%		-30.05%		16.99%	

weight of the penalty function, whereas in **TDCK-Means**  $\alpha$  is the fine-tuning parameter.

**Obtained results.** All the parameters are determined as shown in Section 4.5. Table 1 shows the average values for the indicators, as well as the average standard deviation (in italic) obtained by each algorithm over all values of  $m$ . The average standard deviation is only used to give an idea of the order of magnitude of the stability of each algorithm. Since Simple K-Means, Temporal-Driven K-Means and Constrained K-Means are designed to optimize mainly one component, it is not surprising that they show the best scores for, respectively, the multidimensional variance, the temporal variance and the entropy (best results in boldface). TDCK-Means seeks to provide a compromise, obtaining in two out of three cases the second best score. It is noteworthy that the proposed temporal-aware dissimilarity measure used in Temporal-Driven K-Means provides the highest stability (the lowest average standard deviation) for all indicators. Meanwhile, the constrained algorithms (Constrained K-Means and tcK-Means) show high instability, especially on *Tvar*. TDCK-Means shows a very good stability. The second part of Table 1 gives the relative gain of performance of each of the proposed algorithms over Simple K-Means. It is noteworthy the effectiveness of the temporal-aware dissimilarity measure proposed in Section 3.1, with a 58% gain of Temporal Variance and less than 2% loss of multidimensional variance. The proposed dissimilarity measure greatly enhances the temporal cohesion of the resulted clusters, without a significant scattering of observations in the multidimensional space. Similarly, the Constrained KM shows an improvement in the contiguity measure *ShaP* of 42%, while

Fig. 10. *MDvar* and *ShaP* function of  $\beta$  (a) and of  $\delta$  (b)

losing 10% multidimensional variance. By comparison, tcK-Means shows modest results, improving *ShaP* by only 17% and still showing important losses on both *Tvar* (-30%) and *MDvar* (-2%). This proves that the threshold penalty function proposed in literature has lower performances than our newly proposed contiguity penalty function. TDCK-Means combines the advantages of the other two algorithms, providing an important gain of 43% of temporal variance and increasing the *ShaP* measure by more than 4%. Nonetheless, it shows a 6% loss of *MDvar*.

**Varying the number of clusters** Similar conclusions can be drawn when varying the number of clusters. *MDvar* (Figure 9a) decreases, for all algorithms, as the number of cluster increases. It is well known in clustering literature that the intra-cluster variance decreases steadily with the increase of number of clusters. As the number of clusters augments, so does the differences of TDCK-Means and Constrained K-Means, when compared to the Simple K-Means algorithm. This is due to the fact that the constraints do not let too many clusters to be assigned to the same entity, resulting in the convergence towards a local optimum, with a higher value of *MDvar*. An opposite behavior is shown by the *ShaP* measure in Figure 9c, which increases with the number of clusters. It is interesting to observe how the *MDvar* and the *ShaP* measures have almost opposite behaviors. An algorithm that shows the best performances on one of the measures, also shows the worst on the other. The temporal divergence in Figure 9b shows a very sharp decrease for a low number of clusters, and afterwards remains relatively constant.

#### 4.5. Impact of parameters $\beta$ and $\delta$

The  $\beta$  parameter controls the impact of the contiguity constraints in Equation (4). When set to zero, no constraints are imposed, and the algorithm behaves just like the Simple K-Means. The higher the values of  $\beta$ , the higher the penalty inflicted when breaking a constraint. When  $\beta$  is set to large values, the penalty factor will take precedence over the similarity measure in the objective function. Observations that belong to a certain entity will be assigned to the same cluster, regardless of their resemblance in the description space. When this happens, the algorithm cannot create partitions with higher number of clusters than the number of entities. In order to evaluate the influence of parameter  $\beta$ , we

execute the Constrained K-Means algorithm with  $\beta$  varying from 0 to 0.017 with a step of 0.0005. The value of  $\delta$  is set at 3, and 5 clusters are constructed. For each value of  $\beta$ , we executed 10 times the algorithm and we plot the average obtained values. Figure 10a shows the evolution of measures  $MDvar$  and  $ShaP$  with  $\beta$ . When  $\beta = 0$  both  $MDvar$  and  $ShaP$  have the same values as for Simple K-Means. As  $\beta$  increases, so does the penalty for non-contiguous segmentation of entities.  $MDvar$  starts to increase rapidly, while  $ShaP$  decreases rapidly. Once  $\beta$  reaches higher values, the measures continue their evolution, but with a leaner slope. In the extreme case, in which all observations are assigned to the same cluster regardless of their similarity, the  $ShaP$  measure will reach zero.

The  $\delta$  parameter controls the width of the penalty function in Equation (4). As Figure 3 shows, when  $\delta$  has a low value, a penalty is inflicted only if the time difference of a pair of observations is small. As the time difference increases, the function quickly converges to zero. As  $\delta$  increases, the function decreases with a leaner slope, thus also taking into account observations which are farther away in time. In order to analyze the behavior of the penalty function when varying  $\delta$ , we have executed the Constrained K-Means, with  $\delta$  ranging from 0.1 to 8, using a step of 0.1.  $\beta$  was set at 0.003 and 10 clusters were constructed. Figure 10b plots the evolution of measures  $MDvar$  and  $ShaP$  with  $\delta$ . The contiguity measure  $ShaP$  decreases almost linearly as  $\delta$  increases, as the series of observations belonging to each entity gets segmented in larger chunks. At the same time, the multidimensional variance  $MDvar$  increases linearly with  $\delta$ . Clusters become more heterogeneous and variance increases, as observations get assigned to clusters based on their membership to an entity, rather than their descriptive similarities.

Varying  $\alpha^*$  and  $d^*$  for the tcK-Means proposed in <sup>12</sup> yields similar results, with the  $MDvar$  augmenting and the  $ShaP$  descending, when  $\alpha^*$  and  $d^*$  increase. For the tcK-Means, these evolutions are linear, whereas for the Constrained K-Means they are exponential, following a trend line of function  $e^{-\frac{const}{x}}$ . Plotting the evolution of the  $MDvar$  and the  $ShaP$  indicators on the same graphic, provides a heuristic for choosing the optimum values for the  $(\beta, \delta)$  parameters of the Constrained K-Means and the TDCK-Means, respectively the  $(\alpha^*, d^*)$  parameters of the tcK-Means. Both curves are plotted with the vertical axis scaled to the interval  $[min_{value}, max_{value}]$ . Their point of intersection determines the values of the parameters (as shown in Figure 10a and 10b). The disadvantage of such a heuristic would be that a large number of executions must be performed with multiple values for the parameters before the “optimum” can be found.

#### 4.6. The tuning parameter $\alpha$

The parameter  $\alpha$ , proposed in Section 3.4, allows the fine tuning of the ratio between the multidimensional component and the temporal component in the temporal-aware dissimilarity measure. When  $\alpha$  is close to -1, the temporal component is predominant. Conversely, when  $\alpha$  is close to 1, the multidimensional component takes precedence. The two components have equal weights when  $\alpha = 0$ . To evaluate the influence of parameter  $\alpha$ , we executed Temporal-Driven K-Means with  $\alpha$  varying from -1 to 1 with a step of 0.1. In order not to bias the results and to evaluate only the impact of the tuning parameter, we remove the contiguity constraints from the objective function  $\mathcal{J}$ , by setting  $\beta = 0$ . For

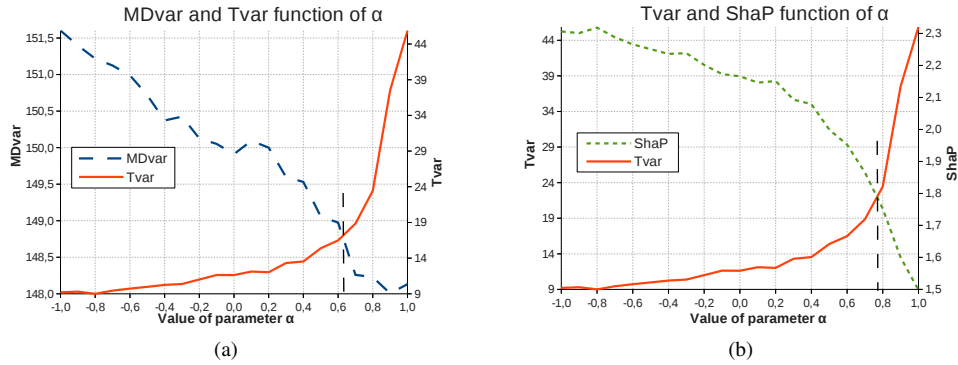


Fig. 11. Influence of tuning parameter  $\alpha$  on *MDvar* and *Tvar* (a) and *Tvar* and *ShaP* (b)

each value of  $\alpha$ , we executed 10 times and we present the average values.

Figure 11a shows the evolution of measures *MDvar* and *Tvar* with  $\alpha$ . For low values of  $\alpha$ , the value of the temporal-aware dissimilarity measure is given mainly by the temporal component, so *Tvar* shows its lowest value, while *MDvar* presents its maximum. As  $\alpha$  increases, *MDvar* decreases as more importance is given to the multidimensional component. For  $\alpha \in (-1, 0]$ , the importance of the temporal component remains intact, the increase of *Tvar* is solely the result of the algorithm converging to a local optimum which also takes into account the multidimensional component. For  $\alpha \in [0, 1)$ , the impact of the multidimensional component stays constant, whereas the importance of the temporal components diminishes. As a result, *MDvar* continues its decrease and *Tvar* increases sharply. For  $\alpha = 1$  the temporal component is basically ignored from the measure. The Temporal-Driven K-Means behaves just like Simple K-Means. Figure 11b shows the evolution of *ShaP* alongside *MDvar*. Even if the contiguity penalty component was neutralized by setting  $\beta = 0$ , the value of *ShaP* is not constant, but it descends with  $\alpha$ . For low values of  $\alpha$ , the temporal component is predominant in the similarity measure. This generates partitions where every cluster regroups all observations from a specific period, regardless of their multidimensional description. This means that all entities have segments in all the clusters, which leads to a high value of *ShaP*.

It is noteworthy that the evolution of the indicators is not linear with  $\alpha$ . As  $\alpha$  increases, *Tvar* augments only very slowly and picks up the pace only for large values of  $\alpha$ . This indicates that the temporal component has an inherent advantage over the multidimensional one. As we presumed in Section 3.4, this is due to the intrinsic nature of the dataset and the main reason why the tuning parameter  $\alpha$  was introduced. The distributions of observations in the multidimensional and temporal spaces is different: in the temporal space, the observations tend to be evenly distributed, whereas in the multidimensional description space, they cluster together. To quantify this, we calculate the ratio between average standard deviation and average distances between pairs of observations:

$$r_{dim} = \frac{1}{|\mathcal{X}|} \sum_{i=1}^{|\mathcal{X}|} \frac{stdev(\{|x_i^{dim} - x_j^{dim}|^2 | x_j \in \mathcal{X}, i \neq j\})}{\frac{1}{|\mathcal{X}|} \sum_{j=1, j \neq i}^{|\mathcal{X}|} |x_i^{dim} - x_j^{dim}|^2}$$

where  $dim$  is replaced with  $d$  or  $t$  (the descriptive or the temporal dimension). On *Comparative Political Data Set I*,  $r_d = 29.5\%$  and  $r_t = 65.3\%$ . This shows that observations are a lot more dispersed in the temporal space than in the multidimensional description space. This explains why  $Tvar$  augments very slowly with  $\alpha$  and starts to increase more rapidly only starting from  $\alpha = 0.4$ .

Following the heuristic proposed in Section 4.5, we can determine a trade-off value for  $\alpha$ . As shown in Figure 11, all vertical axes are magnified between their functions' minimum and maximum values. The trade-off value for  $\alpha$  is found at the intersection point of  $MDvar$  and  $Tvar$  (and  $MDvar$  and  $Shap$ ). This value is set around 0.7, showing the dataset's bias towards the temporal component. This technique for setting the value of the tuning parameter is just a heuristic, the actual value of  $\alpha$  is dependent on the dataset. This is why we are currently working on a method, inspired from multi-objective optimization using evolutionary algorithms<sup>20</sup> to automatically determine the values of  $\alpha$ , as well as the other parameters of TDCK-Means ( $\beta$ ,  $\delta$  and  $\gamma$ ).

## 5. Conclusion and future work

In this paper, we have studied the detection of typical evolutions from a collection of observations. We have proposed a novel way to introduce temporal information directly into the dissimilarity measure, weighting the Euclidean component in the description space by the temporal component. We have proposed TDCK-Means, an extension of K-Means, which uses the temporal-aware dissimilarity measure and a new objective function which takes into consideration the temporal dimension. We use a penalty factor to make sure that the observation series related to an entity get segmented into continuous chunks. We infer a new centroid update formula, where elements distant in time contribute less to the centroid than the temporally close ones. We have proposed an intersection similarity measure between two temporal clusters and a method to calculate *a posteriori* an adjacency matrix. We use this adjacency matrix in order to structure the detected evolution phases as a graph. From a qualitative point of view, we have shown that our algorithm is capable of detecting comprehensible evolutions based on a Political Science dataset. Quantitatively, we have shown that our proposition consistently improves temporal variance, without any significant losses in the multidimensional variance.

We are currently experimenting with applying the algorithm to other applications, *e.g.*, detection of social roles in social networks, by passing through temporal behavioral roles. A social role is defined as a typical succession of behavioral roles. In our current work, we have inferred a temporal cluster graph structure *a posteriori* to the construction of the clusters. We have ongoing work toward incorporating the graph construction into the TDCK-Means algorithm, by modifying the objective function in order to take into account the intersection similarity measure and a temporal distance. Another direction of research will be describing the clusters in a human readable form. We work on means to provide them with an easily comprehensible description by introducing temporal information into an unsupervised feature construction algorithm. We are also experimenting a method for setting automatically the values of TDCK-Means's parameters ( $\alpha$ ,  $\beta$ ,  $\delta$  and  $\gamma$ ), by using an approach inspired from multi-objective optimization using evolutionary algorithms<sup>20</sup>.

## References

1. Klaus Armingeon, David Weisstanner, Sarah Engler, Panajotis Potolidis, Marlène Gerber, and Philipp Leimgruber. Comparative political data set 1960-2009. Institute of Political Science, University of Berne., 2011.
2. Sugato Basu, Arindam Banerjee, and Raymond J. Mooney. Semi-supervised clustering by seeding. In *International Conference on Machine Learning*, pages 19–26, 2002.
3. Mikhail Bilenko and Raymond J. Mooney. Adaptive duplicate detection using learnable string similarity measures. In *Knowledge Discovery and Data Mining, Proceedings of the ninth ACM SIGKDD international conference on*, pages 39–48. ACM New York, NY, USA, 2003.
4. Shixi Chen, Haixun Wang, and Shuigeng Zhou. Concept clustering of evolving data. In *Data Engineering, 2009. ICDE'09. IEEE 25th International Conference on*, pages 1327–1330, 2009.
5. David Cohn, Rich Caruana, and Andrew McCallum. Semi-supervised clustering with user feedback. In *Constrained Clustering: Advances in Algorithms, Theory, and Applications*, volume 4, pages 17–32. Cornell University, 2003.
6. Fernando De la Torre and Carlos Agell. Multimodal diaries. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 839–842. IEEE, 2007.
7. Ayhan Demiriz, Kristin Bennett, and Mark J. Embrechts. Semi-supervised clustering using genetic algorithms. In *Artificial Neural Networks in Engineering*, pages 809–814, 1999.
8. Brigitte Dormont. Petite apologie des données de panel. *Économie & prévision*, 87(1):19–32, 1989.
9. Nizar Grira, Michel Crucianu, and Nozha Boujemaa. Unsupervised and semi-supervised clustering: a brief survey. Technical report, A Review of Machine Learning Techniques for Processing Multimedia Content, Report of the MUSCLE European Network of Excellence (FP6), 2005.
10. Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2):107–145, 2001.
11. Dan Klein, Sepandar D. Kamvar, and Christopher D. Manning. From instance-level constraints to space-level constraints: Making the most of prior knowledge in data clustering. In *International Conference on Machine Learning*, pages 307–314, 2002.
12. Wei-Hao Lin and Er Hauptmann. Structuring continuous video recordings of everyday life using time-constrained clustering. In *IS&T/SPIE Symposium on Electronic Imaging*, 2006.
13. James MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Cam and J. Neyman, editors, *Berkeley Symposium on Mathematical Statistics and Probability, Proceedings of the Fifth*, volume 1, pages 281–297, 1967.
14. Arun Qamra, Belle Tseng, and Edward Y. Chang. Mining blog stories using community-based and temporal clustering. In *Information and Knowledge Management, Proceedings of the 15th ACM international conference on*, pages 58–67, New York, NY, USA, 2006. ACM.
15. Brandon C. S. Sanders and Rahul Sukthankar. Unsupervised discovery of objects using temporal coherence. Technical report, CVPR Technical Sketch, 2001.
16. Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. Coupled temporal scoping of relational facts. In *WSDM*, pages 73–82, 2012.
17. Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schroedl. Constrained k-means clustering with background knowledge. In *International Conference on Machine Learning, Proceedings of the Eighteenth*, pages 577–584. Morgan Kaufmann, 2001.
18. Eric D. Widmer and Gilbert Ritschard. The de-standardization of the life course: Are men and women equal? *Advances in Life Course Research*, 14(1-2):28–39, 2009.
19. Eric P. Xing, Andrew Y. Ng, Michael I. Jordan, and Stuart Russell. Distance metric learning with application to clustering with side-information. *Advances in Neural Information Processing Systems*, 15:505–512, 2002.
20. Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *Evolutionary Computation, IEEE Transactions on*, 11(6):712–731, 2007.